

Tutorial 4: SQL

Informatics 1 Data & Analysis

Week 6, Semester 2, 2014–2015

This worksheet has three parts: tutorial *Questions*, followed by some *Examples* and their *Solutions*.

- Before your tutorial, work through and attempt all of the Questions in the first section. If you get stuck or need help then ask a question on *Piazza*.
- The Examples are there for additional preparation, practice, and revision.
- Use the Solutions to check your answers, and read about possible alternatives.

You must bring your answers to the main questions along to your tutorial. You will need to be able to show these to your tutor, and may be exchanging them with other students, so it is best to have them printed out on paper.

If you cannot do some questions, write down what it is that you find challenging and use this to ask your tutor in the meeting.

Tutorials will not usually cover the Examples, but if you have any questions about those then write them down and ask your tutor, or go along to InfBASE during the week.

It's important both for your learning and other students in the group that you come to tutorials properly prepared. If you have not attempted the main tutorial questions, then you may be sent away from the tutorial to do them elsewhere.

Some exercise sheets contain material marked with a star ★. These are optional extensions.

Data & Analysis tutorials are not formally assessed, but they are a compulsory and important part of the course. If you do not do the exercises then you are unlikely to pass the exam.

Attendance at tutorials is obligatory: if you are ill or otherwise unable to attend one week then email your tutor, and if possible attend another tutorial group in the same week.

Please send any corrections and suggestions to Ian.Stark@ed.ac.uk

Introduction

In the previous tutorial you constructed some queries around a set of relational tables dealing with air travel. In this tutorial, you will formulate queries in SQL for the same application domain, with similar (but not identical) tables. You will also interact with a database using *LibreOffice Base* to execute these queries.

LibreOffice Base

In this tutorial, you will be using *LibreOffice Base*. To set up the program on a DICE machine, you need to follow these steps:

- (a) Download `travel.odt` from the course web page: <http://blog.inf.ed.ac.uk/da15/tutorials>
- (b) From the desktop, choose **Applications** -> **Office** -> **LibreOffice Base**

- (c) When the *Database Wizard* window appears, choose **Open an existing database file**.
- (d) Click **<Open>**.
- (e) Locate the `travel.odb` file you downloaded earlier and click **<Open>**.
- (f) In the *Database* column on the left-hand side, click **Tables**.

You can now select individual tables to inspect their content. To start from the command line instead, enter `libreoffice travel.odb` in a terminal window at the appropriate directory.

The air travel data is now loaded and you are ready to start the tutorial. Make sure you regularly save any data you wish to keep, and print out results to bring to tutorials.

If you wish to try the tutorial on a non-DICE machine then you may want to install your own copy of *LibreOffice* from <http://www.libreoffice.org>. Versions are available for Linux, Windows and Mac. You can also try the very similar *OpenOffice Base* on Windows machines in the open access computer labs (not the ones in the AT café area, though).

SQL Details

Recall the following SQL syntax points from lectures:

- SQL keywords are not case sensitive; identifiers like table or field names may or may not be. Keywords are often written in upper case, and it is good practice to maintain a consistent approach to the cases used in table and field names.
- SQL keywords never contain spaces and never require quotation; SQL identifiers can be quoted using double quotation marks "like this" and if quoted can contain spaces.
- Strings in SQL usually are case sensitive, and must always be quoted, using single quotation marks 'like this'.

Question 1: Queries in SQL

The tables in this database have the same design as the last tutorial. There are four tables — **Airport**, **Booking**, **Flight** and **Seat**. Below you will find a series of English-language queries about this data; for each you should do the following:

- Formulate the specified query in SQL;
- Run the query on the `travel` database;
- Print out the SQL query and the result table.

To run the query on the `travel` database, proceed as follows.

- Click on **Queries** in the left column on the `travel`-window.
- From the list of tasks shown, choose *Create Query in SQL View*.
- Enter your query in the white space in the window that opens.
- Save the query (Ctrl+S) as with the name of your choice.
- Go back to the `travel`-window and double-click on the query name. Results will be displayed in a new window.
- To edit any query, right click on it in the `travel`-window and choose *Edit in SQL View*.

To print the result of a query, open a new text document with *LibreOffice Writer*, and drag the query from the `travel` window into the new document. You'll be given the option to choose some columns: it's simplest to click the `>>` button to keep all of them.

To print the SQL source of a query, edit it in SQL view then select all text and again drag it to a *LibreOffice Writer* document, or the editor of your choice.

When editing your SQL source, you may notice that *LibreOffice* has capitalised keywords and generously added quotation marks around every identifier in sight.

Queries

- (a) Retrieve all the rows in `Airport` table for all the airports in London. The schema of the output table should be same as that of the `Airport` table.
- (b) Retrieve all bookings by British and French passengers. The schema of the output table should be same as that of the `Booking` table.
- (c) Retrieve the names of all the passengers.
- (d) Retrieve the flight number with departure and arrival airports of all British Airways flights.
- (e) Retrieve the name of every passenger together with their flight number and the associated flight company.
- (f) Retrieve all flights from all airports in London. The output schema should be same as that of the `Flight` table.
- (g) Retrieve the ticket numbers and names of all passengers departing from London.
- (h) Retrieve the flight number and flight company of all flights from London to Paris.
- ★ (i) Retrieve the ticket numbers and names of all passengers travelling in Business class.
- ★ (j) Retrieve the names and nationalities of all the Business class passengers travelling from London to Paris.

Question 2: Discussion

So far in this course you have used Relational Algebra, Tuple-Relational Calculus and SQL to formulate queries on tables. From your experience, is SQL more similar to Relational Algebra or Tuple-Relational Calculus? Why have a purpose-built practical query language such as SQL rather than use a theoretically clean language such as Relational Algebra or Tuple-Relational Calculus?

★ Question 3: Extension

The *LibreOffice Base* tool has a graphical user interface to generate and execute queries, with its *Design View* and query creation *Wizard*.

Explore the use of this GUI to create queries; in particular, all of the queries presented earlier. Look at the SQL generated, and compare it to what you created by hand.

Examples

This section contains some examples of SQL queries designed to answer English-language questions. The examples are very similar to the main tutorial questions, but using a database of information about films rather than air travel.

Download the database file `film.odb` from the course web page: <http://blog.inf.ed.ac.uk/da15/tutorials>. You will notice that `film.odb` contains similar (but not identical) tables to those presented in the *Examples* section of Tutorial 3. There are four tables — Actor, Film, Performance and Director. For each of the English-language queries below, questions do the following:

- Formulate the specified query in SQL;
- Run the query on the `films` database;
- Print out the SQL query and the result table.

See the instructions from Question 1 for more detail on how to enter and edit SQL queries in *Libre-Office*.

Queries

- Retrieve details of all films that were released in 2010. The output schema should be the same as that of the Film table.
- Retrieve details of all actors that are not in their thirties. The output schema should be the same as that of the Actor table.
- Retrieve the names of all directors.
- Retrieve the names of all American directors.
- Find out the names of all British actors above the age of 40.
- List all performances by an actor, giving for each one the actor's name and the film in which he or she appeared.
- Find out the names of all actors that have played the character of Bruce Wayne (Batman).
- Retrieve the names of all actors that have played the character of Bruce Wayne, together with the year the corresponding films were released.
- Retrieve all actors from the film Inception. The output schema should be the same as that of the Actor table.
- Find out the names of all actors that have performed in a film directed by Christopher Nolan.
- Retrieve the titles of all films in which Leonardo Di Caprio and Kate Winslet have co-acted.
- Assuming that the ids of actors and directors are used consistently across the tables, retrieve details of all actors that have directed a film.

Solutions to Examples

These are not entirely “model” answers; instead, they indicate a possible solution. Remember that not all of these questions will have a single “right” answer. There can be multiple appropriate ways to formulate a query.

If you have difficulties with a particular example, or have trouble following through the solution, please raise this as a question in your tutorial.

- (a) Retrieve details of all films that were released in 2010. The output schema should be the same as that of the Film table.

```
select * from Film where year = 2010
```

filmId	title	year	directorId
DSP10	Despicable Me	2010	PCF97
INC10	Inception	2010	CN345
SHI10	Shutter Island	2010	SCO78

- (b) Retrieve details of all actors that are not in their thirties. The output schema should be the same as that of the Actor table.

```
select * from Actor where age < 30 or age > 39
```

actorId	name	nationality	age
CB379	Christian Bale	British	40
DP423	Dev Patel	British	24
ELP87	Ellen Page	American	27
EMG32	Ewan McGregor	British	43
HBC54	Helena Bonham Carter	British	48
JD801	Judi Dench	British	80
LDC21	Leonardo DiCaprio	American	40
MKE12	Michael Keaton	American	63

- (c) Retrieve the names of all directors.

```
select name from Director
```

name
Darren Aronofsky
Tim Burton
Christopher Nolan
Danny Boyle
James Cameron
Michael Keaton
Pierre Coffin
Martin Scorsese
Sam Mendes

- (d) Retrieve the names of all American directors.

```
select name from Director where nationality = 'American'
```

name
Darren Aronofsky
Tim Burton
Michael Keaton
Martin Scorsese

(e) Find out the names of all British actors above the age of 40.

```
select name from Actor
where nationality = 'British' and age>40
```

name
Ewan McGregor
Helena Bonham Carter
Judi Dench

(f) List all performances by an actor, giving for each one the actor's name and the film in which he or she appeared.

```
select name, title from Actor A, PerformsIn P, Film F
where A.actorId = P.actorId and P.filmId = F.filmId
```

name	title
Christian Bale	The Dark Knight
Christian Bale	The Dark Knight Rises
Dev Patel	Slumdog Millionaire
Ellen Page	Inception
Ewan McGregor	Big Fish
Ewan McGregor	Trainspotting
Helena Bonham Carter	Big Fish
Judi Dench	Skyfall
Joseph Gordon-Levitt	The Dark Knight Rises
Joseph Gordon-Levitt	Inception
Kate Winslet	Revolutionary Road
Kate Winslet	Titanic
Leonardo DiCaprio	Inception
Leonardo DiCaprio	Revolutionary Road
Leonardo DiCaprio	Shutter Island
Leonardo DiCaprio	Titanic
Michael Keaton	Batman Returns

(g) Find out the names of all actors that have played the character of Bruce Wayne (Batman).

```
select distinct name from Actor A, PerformsIn P
where A.actorId = P.actorId and P.character = 'Bruce Wayne'
```

Alternate solutions are possible, using features such as explicit joins or nested queries. These aren't an examinable part of the course, though. For example:

```
select distinct name from Actor
where actorId in (select actorId from PerformsIn where character = 'Bruce Wayne')
```

name
Christian Bale
Michael Keaton

(h) Retrieve the names of all actors that have played the character of Bruce Wayne, together with the year the corresponding films were released.

```
select A.name, F.year from Actor A, PerformsIn P, Film F
where A.actorId = P.actorId and P.character = 'Bruce Wayne' and P.filmId = F.filmId
```

Here is an alternate presentation of this solution using explicit joins — notice that the tables mentioned and the equality tests used are exactly the same, just arranged to show one way the tables could be joined to evaluate this.

```

select A.name, F.year
from Actor A
  join PerformsIn P on A.actorId = P.actorId
  join Film F on P.filmId = F.filmId
where P.character = 'Bruce Wayne'

```

name	year
Christian Bale	2008
Christian Bale	2012
Michael Keaton	1992

- (i) Retrieve all actors from the film Inception. The output schema should be the same as that of the Actor table.

```

select A.* from Actor A, PerformsIn P, Film F
where A.actorId = P.actorId and P.filmId = F.filmId and F.title = 'Inception'

```

actorId	name	nationality	age
ELP87	Ellen Page	American	27
JGL81	Joseph Gordon-Levitt	American	33
LDC21	Leonardo DiCaprio	American	40

- (j) Find out the names of all actors that have performed in a film directed by Christopher Nolan.

```

select distinct A.name from Actor A, PerformsIn P, Film F, Director D
where A.actorId = P.actorId and P.filmId = F.filmId
  and F.directorId = D.directorId and D.name = 'Christopher Nolan'

```

name
Christian Bale
Ellen Page
Joseph Gordon-Levitt
Leonardo DiCaprio

- (k) Retrieve the titles of all films in which Leonardo Di Caprio and Kate Winslet have co-acted.

```

select distinct F.title from Film F, Actor A1, Actor A2, PerformsIn P1, PerformsIn P2
where A1.name = 'Leonardo DiCaprio' and A2.name = 'Kate Winslet'
  and A1.actorId = P1.actorId and P1.filmId = F.filmId
  and A2.actorId = P2.actorId and P2.filmId = F.filmId

```

title
Revolutionary Road
Titanic

- (l) Assuming that the ids of actors and directors are used consistently across the tables, retrieve details of all actors that are also in the database as having directed a film.

```

select A.* from Actor A, Director D
where A.actorId = D.directorId

```

actorId	name	nationality	age
MKE12	Michael Keaton	American	63